Robin Milner and Mads Tofte, *Commentary on Standard ML* (MIT Press, London, 1991), Price $26.95, ISBN 0-262-61327-7 (paperback), Price $40.50, ISBN 0-262-13271-0 (hardcover).

This book is a companion to *The Definition of Standard ML* by the same authors with Robert Harper. In the following they will be called the *Commentary* and the *Definition* for short. The authors tell in the Preface, that the two books play a complementary role—the *Definition* tells *what*, the other *why*. The intended audience is wide: implementers, programmers, and programming language teachers and researchers.

In addition to the Preface, the *Commentary* contains eleven chapters, four appendices and an Index. The chapters fall in four groups: Chapter 1 introduces the machinery of operational semantics; Chapters 2 and 3 are about the dynamic semantics; Chapters 4–8 treat static semantics; and Chapters 9–11 are concerned with the theory of semantic objects for modules.

The organization of the book matches the one in the *Definition* but not on a chapter-to-chapter basis. The comments contain clear references to the *Definition*, and the Index contains entries for the rules of the *Definition*. Several exercises are offered and one of the appendices contains their solutions. Mistakes and ambiguities in the *Definition* are indicated in another appendix.

Reading the book without having the *Definition* ready at hand is senseless. Implementers will no doubt read the *Commentary* cover to cover and benefit from doing so. Programmers and teachers should be warned that reading the book is tough, but they should definitely be aware of its existence. They should familiarize themselves with the technical machinery and learn to use the *Definition* and the *Commentary* as references when in doubt about semantics.

Researchers may want to read the book for various reasons. They may find that the book is just as much an example of using operational semantics as about ML. The chapters on semantic objects for modules will, however, be of particular interest, as they contain detailed proofs of essential theorems on which ML's notion of modules rests. The book will be useful for those who want to investigate language design issues within the limits set by the fundamental design of ML. It will be of little help in the design of languages that are intended to break these—e.g. languages with a different view of the notions module and functor.

Emphasis is overwhelmingly on the technical aspects. As the authors tell in the Preface: "We also point out some cases where the decision was not fully determined; that is, cases where we cannot argue conclusively against an alternative". Of course we need a recording of the detailed arguments that fully determined a decision; but some may find that those decisions that are not forced are particularly interesting, and hence it would have been nice to have a summary of those cases mentioned.

The book is well written and the technical material is presented in a pedagogical way, which should help in not narrowing the actual audience too much. The ML community is fortunate at this stage in the development to find such excellent authors

among the most technically competent in the field. I look forward to using the book regularly, but not intensively, and I strongly recommend the *Commentary* and the *Definition* to those interested in ML who know that such books are not intended for teaching ML nor to increase actual programming skills.

Jørgen STEENSGAARD-MADSEN
Technical University of Denmark
Lyngby, Denmark

G. Huet and G. Plotkin, eds., *Logical Frameworks* (Cambridge University Press, Cambridge, England, 1991), Price £30.00, $65.00 (hardcover), ISBN 0-521-41300-1.

The concept of logical framework applies to formal frames for building various deductive systems. Considering the implementation requirement, it obtains also features of computational environments (of workbenches) for logic. Both aspects are considered in the present book. In building verification systems for software and hardware as well as formal program construction systems one can benefit from logical frameworks as tools. This could be a practical justification of the interest in logical frameworks. However, there is a deeper motivation of research in this field—investigating the computational content of logic. Realizability of logical formulas, observed by Kolmogorov and Heyting in the thirties, was presented by Curry and Howard in the form of a "propositions-as-types" interpretation and developed by Martin Löf in his intuitionistic theory of types. The latter possesses the generality of a framework for implementing various logical theories, but is too hard to "computerize" directly in its full generality. Starting from the first Automath framework developed by de Bruijn more than twenty years ago, the work on logical frameworks began. However, the implementation part of it has remained mostly experimental, whereas interesting results have been obtained in deeper understanding of deductive systems, type theories and their relation to computations. It has become clear that very fundamental ideas of program development originate from here. In a somewhat vulgar way one cay say that Lisp and Prolog (i.e., functional and logic programming of today) are only shallow reflections of this basic research in computational logic which is the content of the book.

This is a collection of carefully selected papers produced as a result of the First Annual Workshop on Logical Frameworks. The papers are grouped into the following six chapters:

- Frameworks,
- Implementations,
- Representing Formal Systems,
- Type Theory,
- Proofs and Computations,
- Logical Issues.